

Trabajo teórico MAC

Curso

2016-2017

Por Néstor Sabater Ballestrín

www.nsabater.com

Objeto

El trabajo consiste en la realización de un script para 3ds Max Studio, en lenguaje maxscript, que permita realizar tareas que a mano serían laboriosas o poco efectivas.

Concepto

El script desarrollado por mi persona, Néstor Sabater, trata de solventar la necesidad de **generación de muros realistas** (por ladrillos) que son utilizables en plugins como massFX para su destrucción de manera procedural a partir de los parámetros que el usuario establezca.

Funcionalidades

En el tiempo disponible se han logrado implementar satisfactoriamente los siguientes parámetros modificables por el usuario:

Muro:

- Extensión del muro (largo x alto x ancho)
- Muro de una sola pieza o de ladrillos sueltos (bool)
- Tamaño de los ladrillos
- Material a aplicar a los ladrillos
- Distorsión cada X ladrillos (gestión de irregularidades)
- Libertad lateral (slider para la generación de las irregularidades)
- Añadir ruido (bool, deforma los ladrillos a efecto de rocas)
- Chamfer, redondea los bordes de los ladrillos para obtener un acabado más realista

Puerta:

- Dimensiones (altura x anchura)
- Posición de la puerta (slider de posición relativa a la longitud del muro)

Iluminación:

- Número de luces
- Offset en Y de las luces
- Offset en Z de las luces
- Intensidad de las luces
- Color de las luces

Funcionamiento

El script ha sido desarrollado teniendo en cuenta diferentes factores.

En las tareas modificación de las dimensiones del muro se ha optado por, en lugar de redimensionar en tiempo real conforme se modifican los parámetros, **mostrar un gizmo a modo de vista previa**, pues hará su uso mucho más rápido y fluido que si se añaden/eliminan ladrillos con sus respectivos modificadores en T real.

En cuanto a los demás parámetros, todos excepto el booleano de añadir ruido a los ladrillos **se pueden modificar en tiempo real** permitiendo ver cómo afectan los cambios que se están realizando.

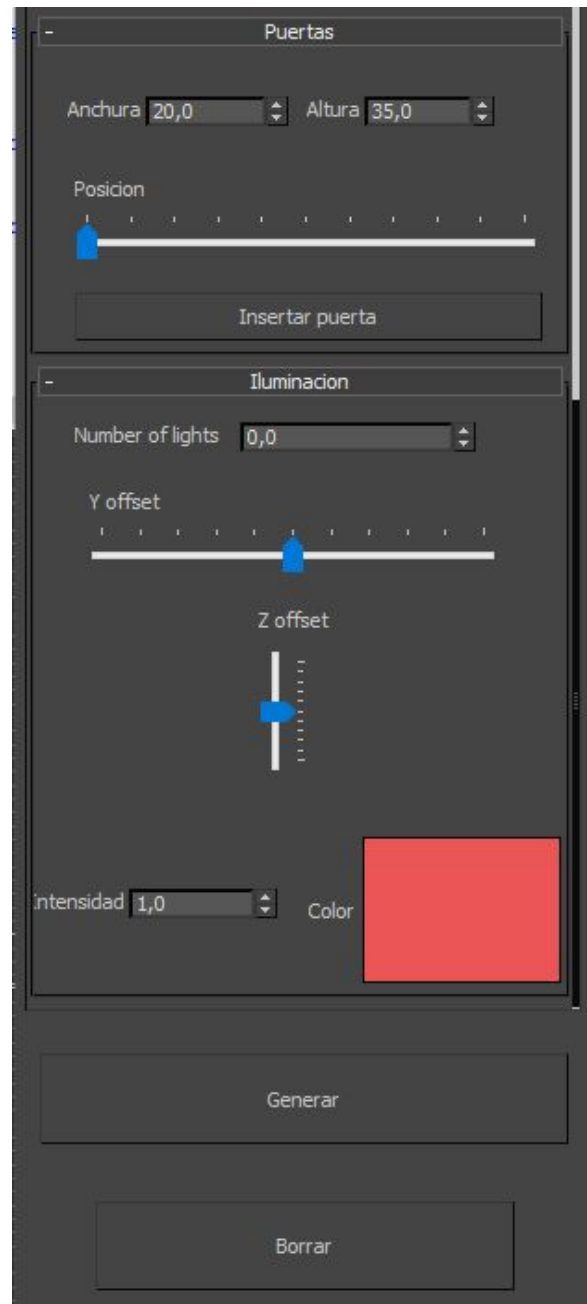
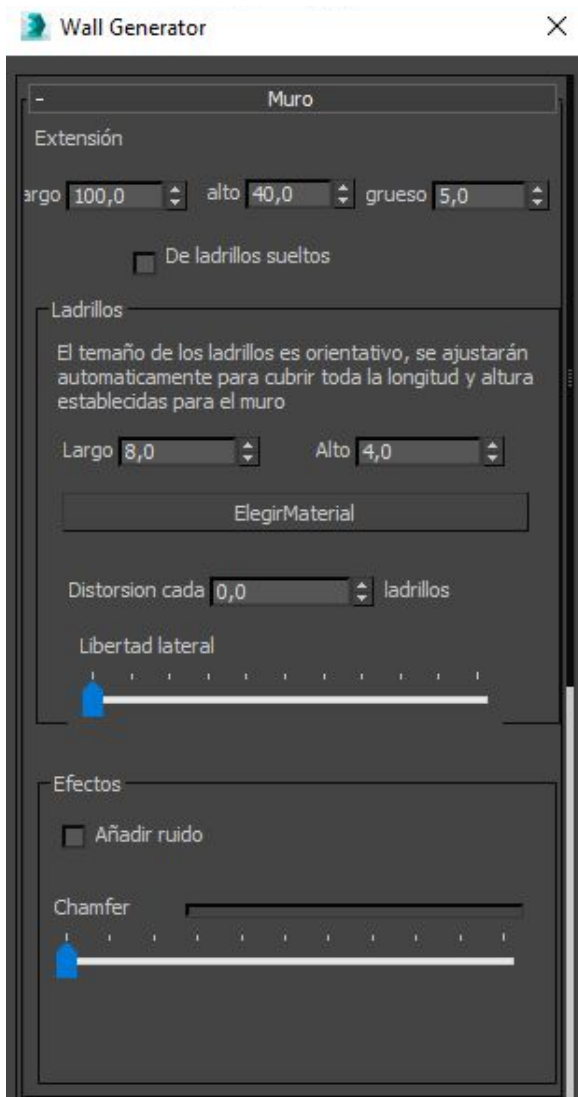
El código

Este script consta de 799 líneas de código en un único archivo, donde se ha tratado de seguir las buenas prácticas recomendadas en la programación.

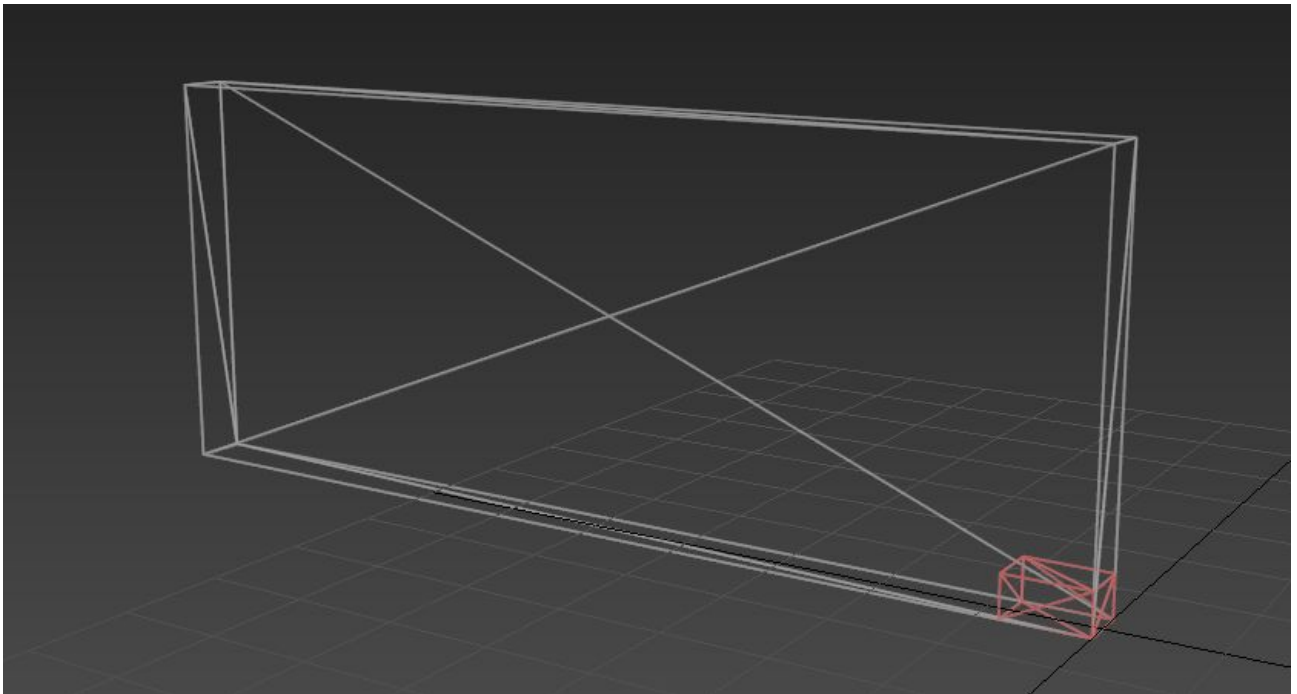
```
1  -- Global vars --
2
3  centroX = 0
4  centroY = 0
5  centroZ = 0
6
7  wallLength = 100
8  wallHeight = 40
9  wallThickness = 5
10
11
12  -- ##### BRICK VARS #####--
13
14  brickLength = 8
15  brickHeight = 4
16
17  -- false == liso, true == ladrillos --
18  useBricks = false
19  addNoise = false -- whether or not to add noise to bricks
20
21  brickChamfer = 0
22  chamferProgress = 0
23
24  brickMaterial = undefined
25
26  allowableBrickDistortionFrequency = 5 -- mininum distance between distortions
27  brickLateralDistortion = 0 -- lateral oscilation for distorted bricks
28
29
30
31  bricksUsed_altitude = 0
32  bricksUsed_longitude = 0
33
34  -- ##### GROUND VARS #####--
35
36  groundMaterial = undefined
37
38  -- ##### LIGHT VARS #####--
39
336 fn applyChamfer obj = (
337   tmpChamfer = chamfer()
338   --tmpChamfer.amount = 0.030
339   --tmpChamfer.amount = 0.30
340   tmpChamfer.amount = brickChamfer
341   tmpChamfer.segments = 1
342   tmpChamfer.tension = 0.69
343
344   addmodifier obj tmpChamfer
345 )
346
347 fn applyNoisemodifier obj = (
348   tmpNoise = Noisemodifier()
349   tmpNoise.seed = random 6 7
350   tmpNoise.scale = 18.107
351   tmpNoise.strength = [units.decodeValue "175.0cm",units.decodeValue "420.0cm",unit
352   tmpNoise.frequency = 0.15
353
354   addmodifier obj tmpNoise
355 )
356
357 fn addModifiers = (
358   --deleteModifier arryModChamfer
359   --deleteModifier arryModNoise
360
361   if arryBricks.count > 0 do(
362     -- this IF only pretends to take less process time for the script
363     if hasModifier arryBricks[1] chamfer == false and brickChamfer > 0 and hasModifi
364       max modify mode
365       for i = 1 to arryBricks.count do(
366         applyChamfer arryBricks[i]
367         applyNoisemodifier arryBricks[i]
368       )
369     )
370   )else(
371     if hasModifier arryBricks[1] chamfer == false and brickChamfer > 0 do(
372       max modify mode
373       for i = 1 to arryBricks.count do(
374         applyChamfer arryBricks[i]
375       )
376     )
377   )
378   if hasModifier arryBricks[1] Noisemodifier == false and addNoise == true do(
379     max modify mode
380   )
```

Diseño de la interfaz

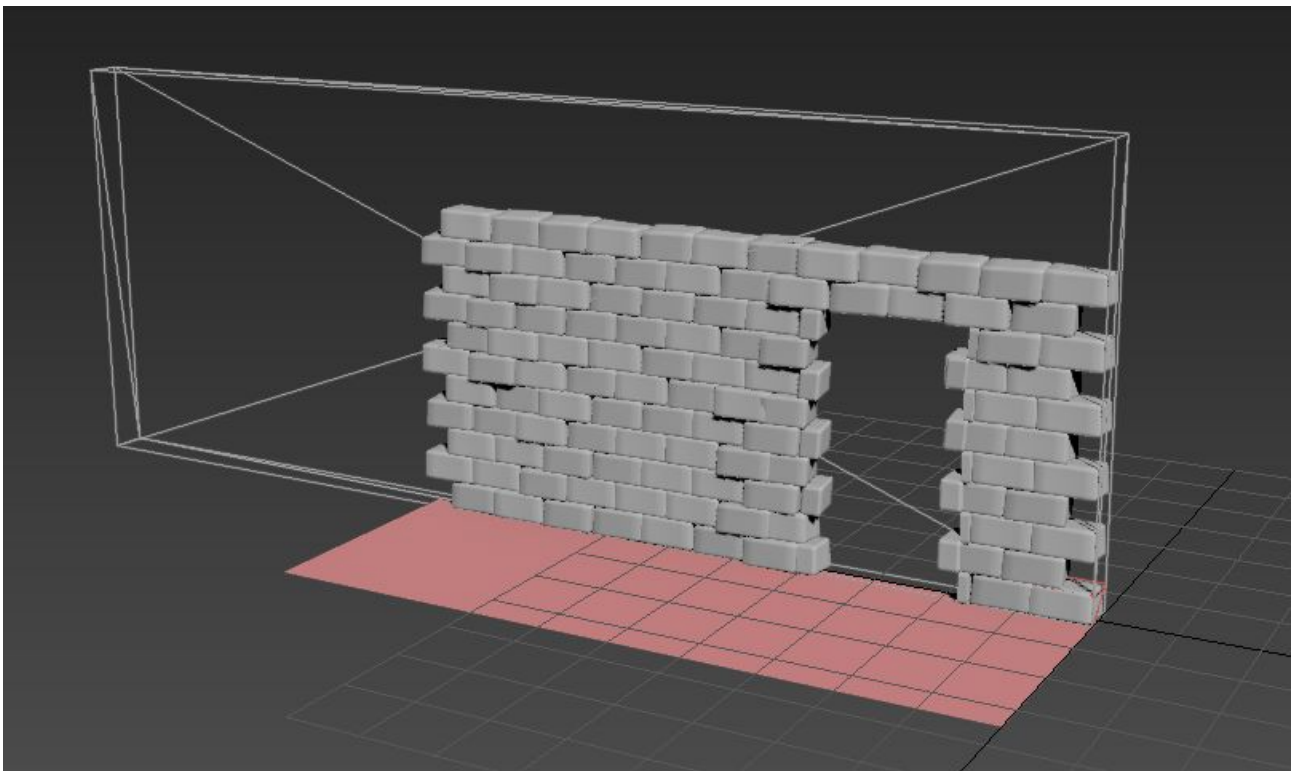
Se ha seguido un diseño acorde a la UI de 3ds Max, esto significa utilizar paneles alargados verticalmente y agrupados en diferentes rollouts.



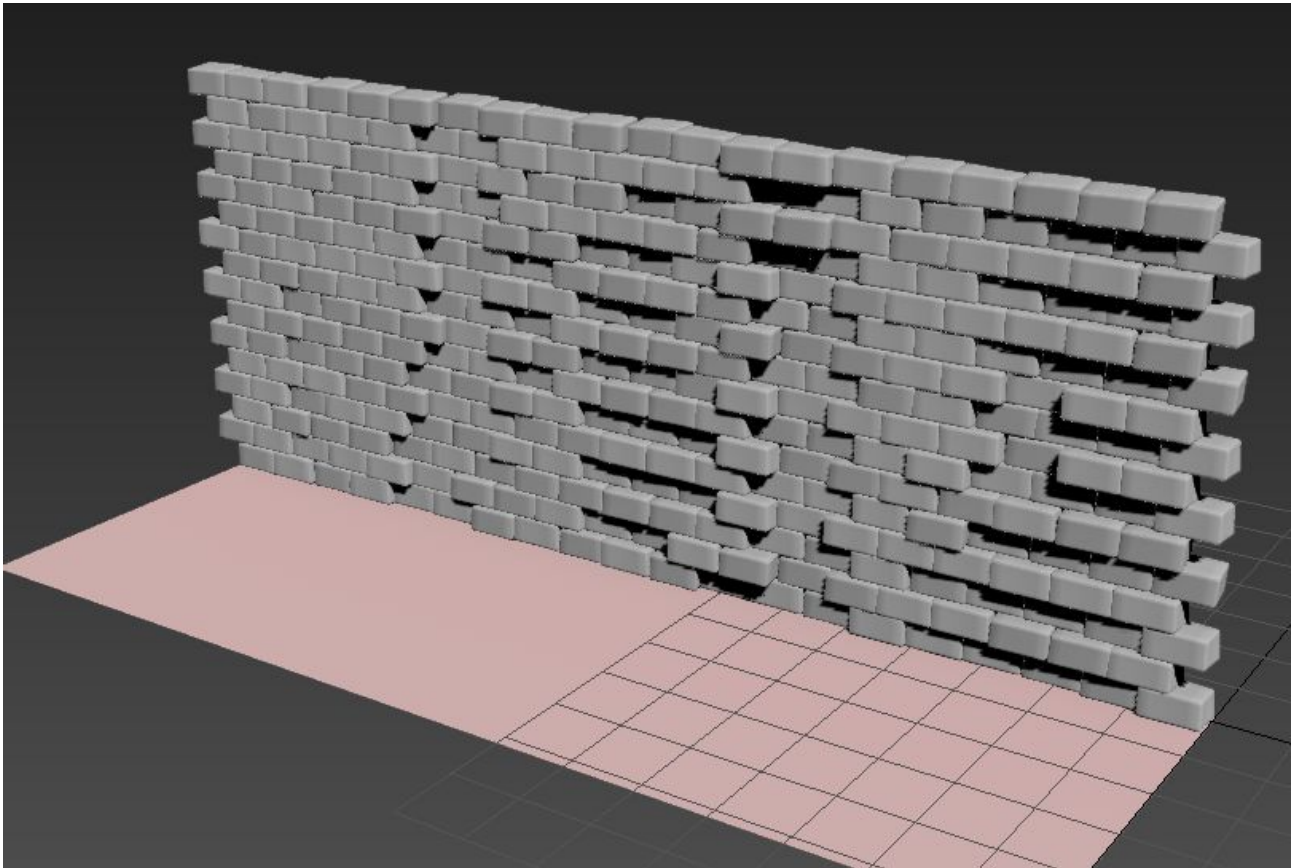
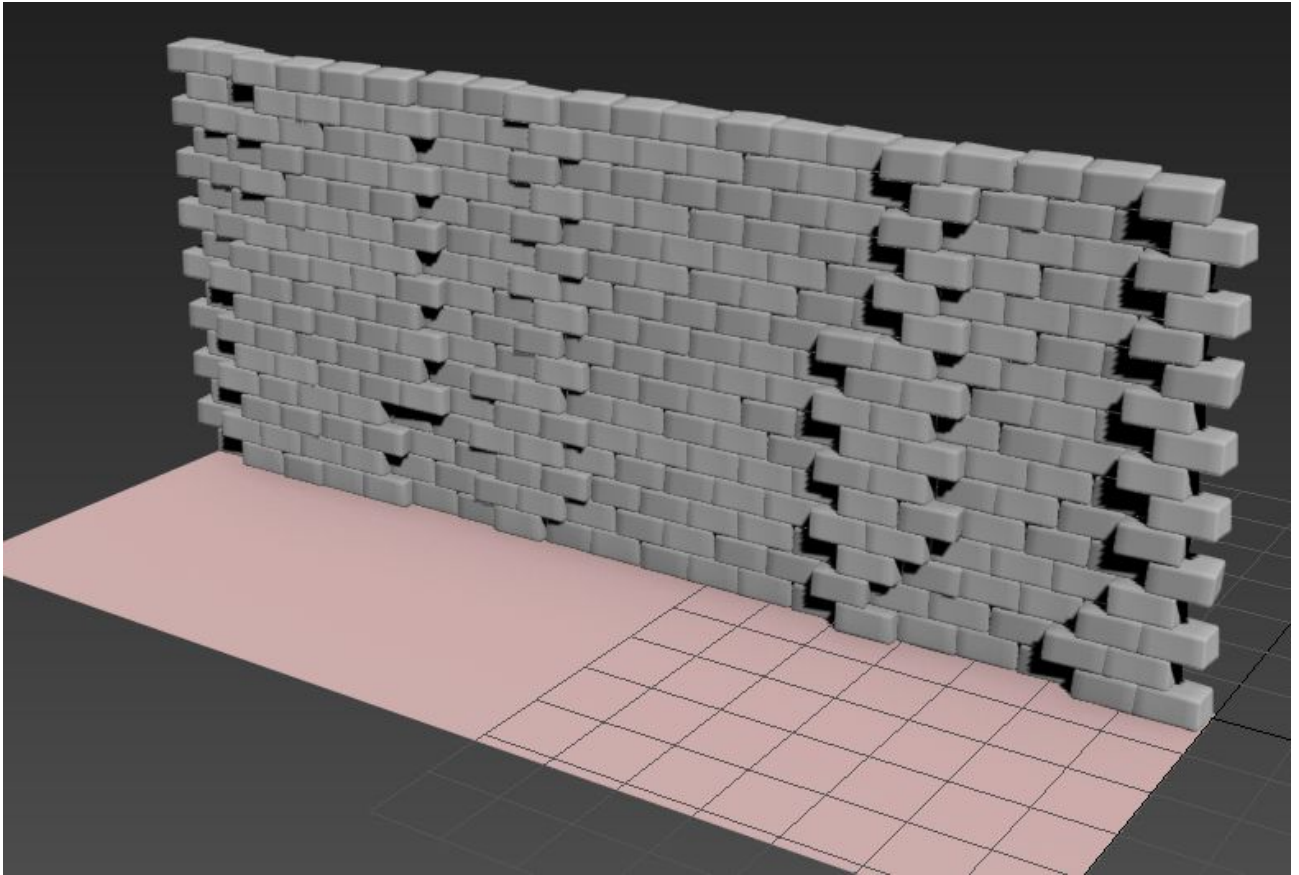
Capturas de uso

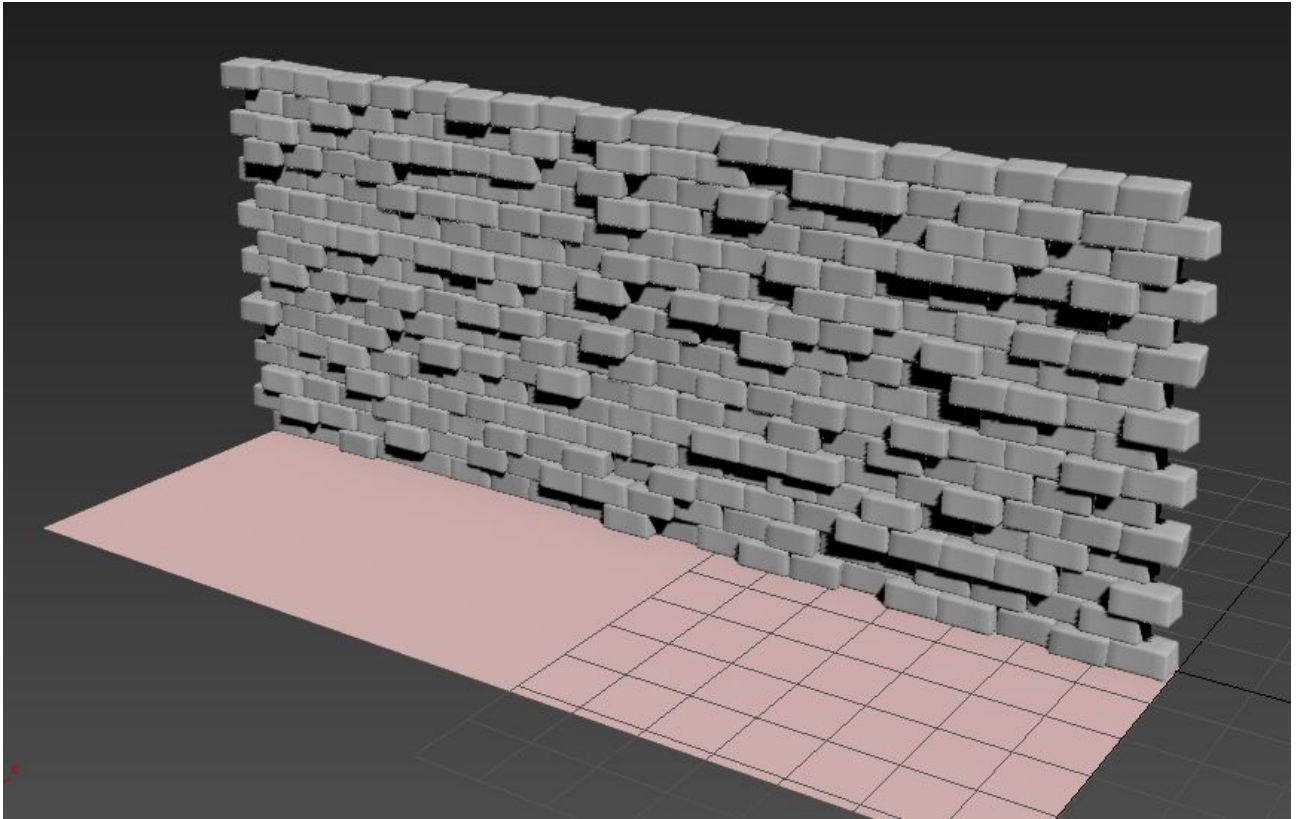


Gizmo de dimensiones del muro y ladrillos

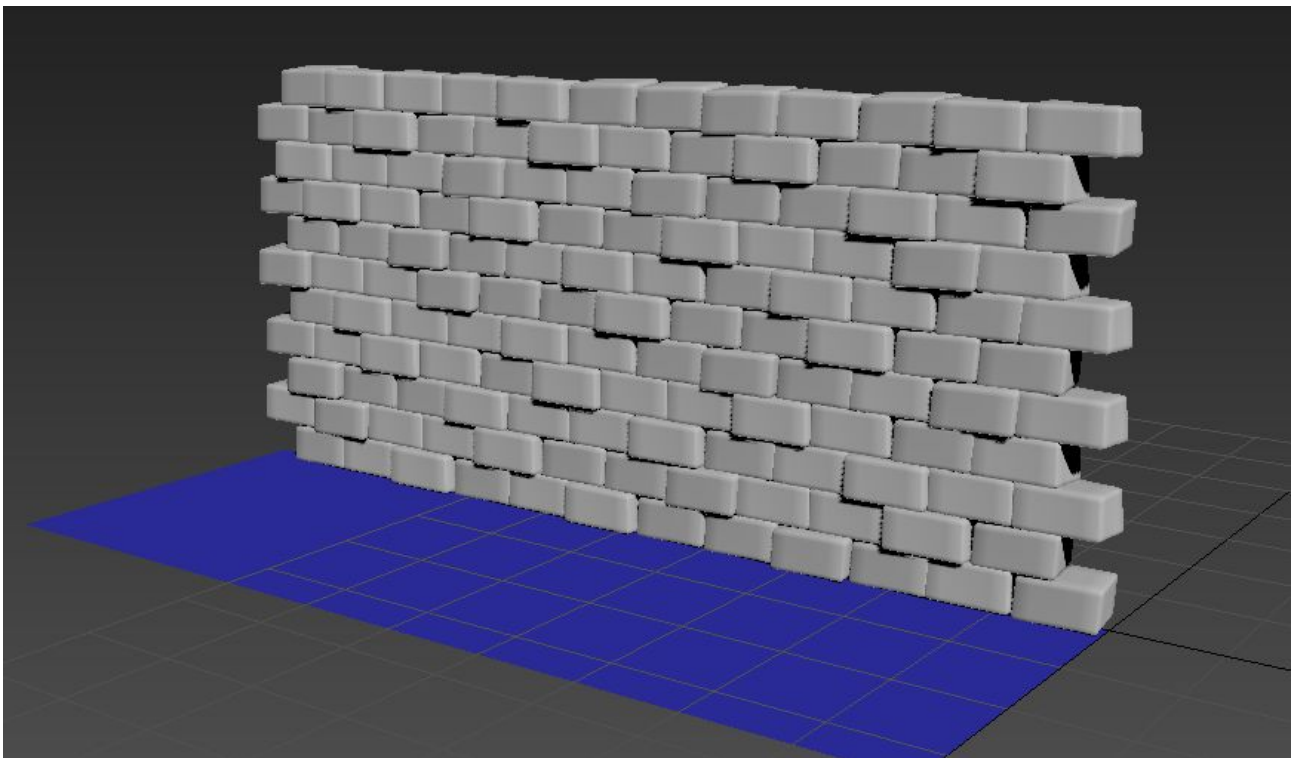


Gizmo de reescalado del muro

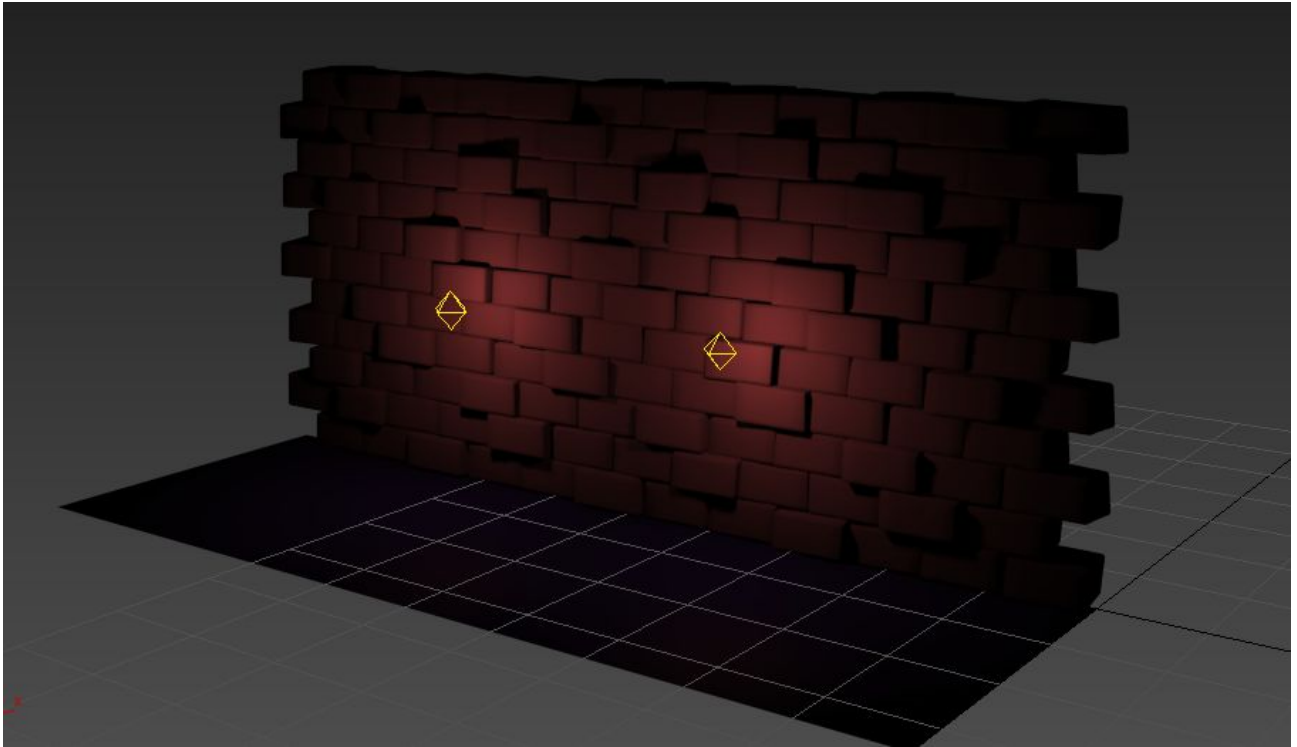




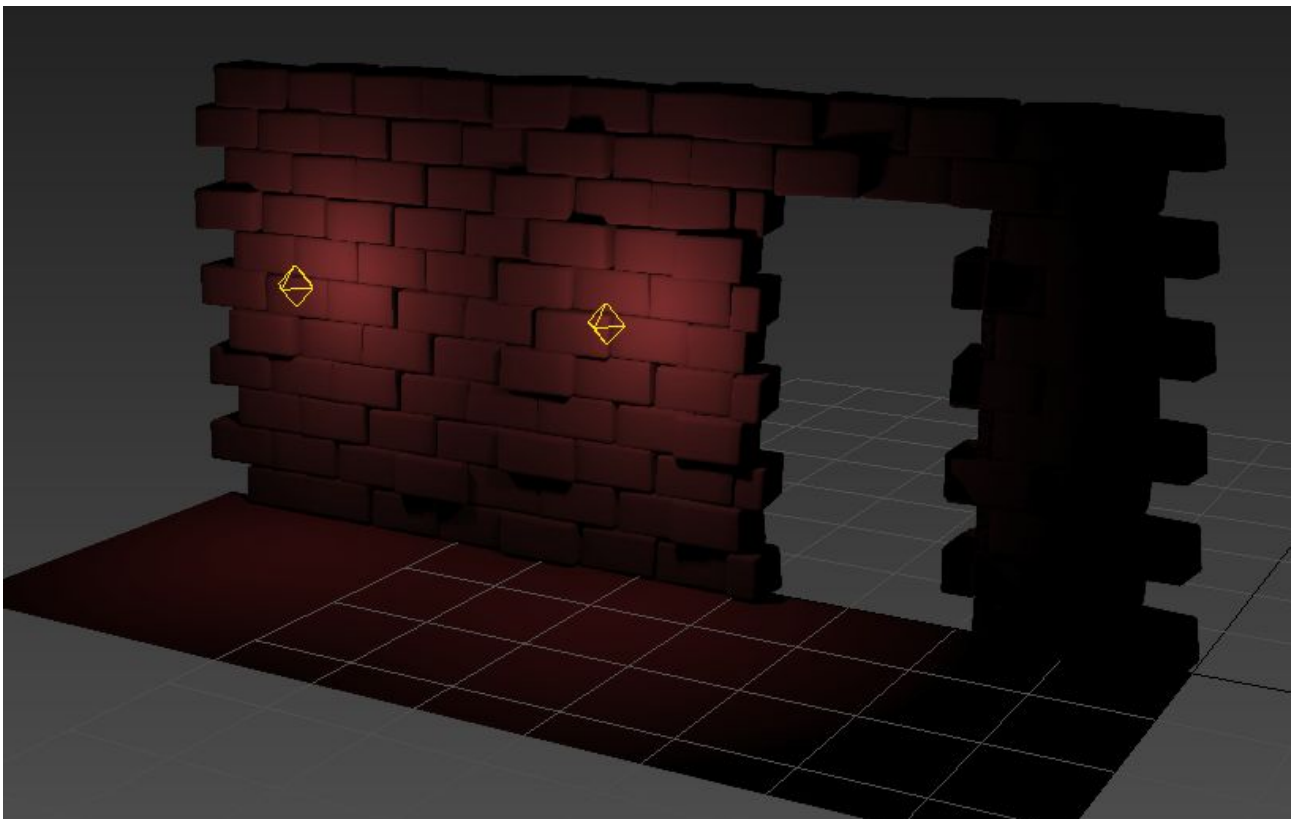
Gracias a los parámetros de distorsión (dis. cada x ladrillos) podemos crear diferentes efectos aleatorios de sobresalido de ladrillos.



Muro de ladrillos, con material, sin luces, con distorsión lateral, chamfer y ruido.



Mismo muro con dos luces



Muro con dos luces desplazadas en Y & Z, mas puerta generada.

Mejoras y ampliaciones

El script será publicado en el portfolío personal www.nsabater.com con lo que se planea completar ligeramente las funcionalidades actuales.

Del mismo modo se implementará la generación de ventanas una vez se solventen pequeños errores en los cálculos de la generación de puertas que en ocasiones producen resultados poco realistas.

Conclusiones

Ha sido **un trabajo interesante** desde el punto de vista didáctico. Nunca había programado en este lenguaje de scripting pero personalmente no me ha parecido diferente de los demás, especialmente después del uso habitual de JavaScript y PHP.

En cuanto al cálculo de posiciones en 3D y transformaciones, la **API de maxscript** es **bastante completa** y te permite abstraerse de ello, preocupándote solo de tus propias ideas y cómputo de vectores posición. No es mi parte favorita, pero eso sí lo había realizado con anterioridad en motores de videojuegos como **Unreal Engine**, Unity 4 o CryEngine